

Research on multi - objective optimization algorithm based on bucket principle

Honggao Wei ^a, Shucheng Huang, Zhicheng Miao

Jiangsu University of Science and Technology, Jiangsu 212003, China.

^a137247136@qq.com

Keywords: Engineering design problem, Multi-objective, Constraint condition, Bucket principle, Genetic algorithm, Optimization algorithm.

Abstract. Most engineering design problems have multiple objectives, under the premise of satisfying constraints, it is necessary to maximize (or minimize) these goals at the same time. Under normal circumstances, multi-target corresponds to a variety of solutions, which usually use genetic algorithm to find its Pareto solution. As the current genetic algorithm to solve the problem of the process is too complicated, this paper uses the method of bucket and genetic algorithm to solve the multi-objective optimization problem considering two targets. By comparing the experimental results, it has achieved good results. The steps of this algorithm are relatively simple, and provides a new way to solve the multi-objective optimization problem under certain constraints.

1. Introduction

In engineering design and other production practice, it often make a plan for the cost, risk, income and other multi-objective evaluation. Trying to find a solution to meet the various objectives, which is not easy. In most cases, the essence of multi-objective optimization is that the improvement of a target, which may cause performance of other target to be reduced. While it is impossible to achieve multiple goals at the same time, the solution of this problem is often troubled by people.

2. Description of Solving Multi - Objective Problems

In order to solve the multi-objective optimization problem, a mathematical model is described.

$$\begin{cases} V - \min f(x) = [f_1(x), f_2(x), \dots, f_p(x)]^T \\ s.t. x \in X \\ X \subseteq R^m \end{cases} \quad (1.1)$$

Which $V - \min$ means that the vector minimization, that is, each sub-objective function is as small as possible in vector target $f(x) = [f_1(x), f_2(x), \dots, f_p(x)]^T$. x is the feasible solution, X is the feasible solution set, R^m is the search space of the solution.

The solution of the above model is different from the single-objective optimization problem. In order to solve the multi-objective optimization problem correctly, we also define the multi-objective optimal solution and the Pareto optimal solution, As defined in definition 1, definition 2, and definition 3:

Definition 1 $X \subseteq R^m$ is a set of constraints for a multi-objective optimization model,

$f(x) \in R^p$ is a vector objective function for multi-objective optimization.

If $\begin{cases} f_k(x_1) \leq f_k(x_2), (\forall k = 1, 2, \dots, p) \\ f_k(x_1) < f_k(x_2), (\exists k = 1, 2, \dots, p) \end{cases}$, Then solution x_1 is better than x_2 .

Definition 2 $X \subseteq R^m$ is constraint sets of multi-objective optimization models and $f(x) \in R^p$ is a vector object for multi-objective optimization. If $x_1 \in X$, and x_1 is superior than all other solutions in X , x_1 is called the optimal solution of the multi-objective optimization model.

Definition 3 $X \subseteq R^m$ is constraint sets of multi-objective optimization models and $f(x) \in R^p$ is a vector object for multi-objective optimization. If $x_1 \in X$, and there is not x_1 which is superior than all other solutions in X , then x called Pareto optimal solution of multi-objective optimization model, or non-inferior solution.

From the definition 2, we can see that the optimal solution of the multi-objective optimization problem is to make the solution of each sub-objective function of the function reach the optimal point at the same time. And the Pareto optimal solution x which satisfy definition 3 is not necessarily unique.

3. Combination of Bucket Principle and Genetic Algorithm Experiments

3.1 The Combination of Bucket Principle and Genetic Algorithm

Genetic algorithms find the optimal solution by imitate biological evolution, which are often used to find the optimal solution in the massive solution. Therefore, genetic algorithm is a common algorithm in the process of multi-objective optimization. The process of genetic algorithm is generally divided into coding, generating population, function evaluation, screening out the parent generation, crossing the generation of offspring, mutating the offspring, and then performing the function evaluation, screening out the parent, cross the generation, making a mutation, go back and forth until you find a satisfactory solution or reach a certain number of iterations. Screening stage is an important stage, if it is a single objective function, more mature screening methods are elite selection method, roulette selection method and so on. However, population under the multi-objective function evaluation, it is difficult to solve the multi-objective optimization problem by using the genetic algorithm because it cannot uniquely determine that the individual is better than the other individual. The multi-objective problem is usually transformed into single-goal problem by weighted method, this method can solve some problems. Because the conversion caused by a single target and multi-objective fact is not equivalent to the loss of practical value.

The core content of the bucket principle is: the number of barrels of water does not depend on the highest piece of wood, but depends on the shortest piece of barrel. According to this content, there can be two corollaries: first, only the walls of the board are all high enough, the barrel can be filled with water; second, as long as the barrel is not high enough, water cannot be full. In other words, to find ways to let the board to reach the height of the board or all the board to maintain equal height of "high enough", which can fully play the role of the team, fully embodies the team spirit. The principle of bucket and the multi-objective problem finding the optimal solution is the same, this paper introduces the principle of bucket to solve this problem: suppose there are p objective functions, followed by the f_1, f_2, \dots, f_p , p function values were used to sort the population, get p sorting. The final one of each sort take as a short barrel of the principle of the barrel. For this p short board, not all abandoned, but randomly on a certain short board by a certain probability to abandon. According to a certain probability to abandon, the reason is that this short board may also contain a useful gene, if directly abandoned, it will soon make the population lost gene, and it does not exist good genes after it is used for some time. In this way to promote the population in all directions to develop as much as possible.

3.2 Experimental Simulation

Design of a scenario to test this algorithm: a factory has 10 plants, each plant has a machine, and there is a certain distance between the plants, there are 6 parts need to be processed, each part need to be processed by 6 processes, and each process has 1 or 2 machines. When choose a different machine, processing time is also different. Constraints are: each part of the process before the end of a process which can carry out the next process, each machine can only accept a part at the same time.

The following list is given for the various data:

Table 1 Machine Selection Table

process1	3, 10	2	3, 9	4	5	2
process 2	1	3	4, 7	1, 9	2, 7	4, 7
process 3	2	5, 8	6, 8	3, 7	3, 10	6, 9
process 4	4, 7	6, 7	1	2, 8	6, 9	1
process 5	6, 8	1	2, 10	5	1	5, 8
process 6	5	4, 10	5	6	4, 8	3

Table 2 Parts Processing Schedule

process 1	3, 5	6	1, 4	7	6	2
process 2	10	8	5, 7	4, 3	10, 12	4, 7
process 3	9	1, 4	5, 6	4, 6	7, 9	6, 9
process 4	5, 4	5, 6	5	3, 5	8, 8	1
process 5	3, 3	3	9, 11	1	5	5, 8
process 6	10	3, 3	1	3	4, 7	3

Table 3 Machine Transfer Fee Table

plant	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0	2	9	7	5	1	6	5	2	1
M2	2	0	8	2	6	1	6	2	5	4
M3	9	8	0	2	8	3	1	1	6	4
M4	7	2	2	0	1	4	2	6	1	8
M5	5	6	8	1	0	6	4	9	6	8
M6	1	1	3	4	6	0	8	5	1	9
M7	6	6	1	2	4	8	0	5	1	4
M8	5	2	1	6	9	5	5	0	5	9
M9	2	5	6	1	6	1	1	5	0	9
M10	1	4	4	8	8	9	4	9	9	0

Now set two objective functions,

(1) $f_1 = \min$ (time of machine)

(2) $f_2 = \min$ (cost of parts)

Set three contrast experiments:

Experiment 1: Find the good solution under the constraint of single objective function f_1 ;

Experiment 2: Find the good solution under the constraint of single objective function f_2 ;

Experiment 3: Find a good solution under constraint of a two - objective function f_1 and f_2

3.3 Experimental description

Experiment 1 and Experiment 2 are the problem of the optimal solution of the traditional single objective function. The purpose is to find out the limit of the function value in the case of single target, which is helpful to evaluate the multi-objective solution under experiment 3. The solution is omitted, giving the results of the experiment. (Note: this solution is not necessarily the optimal solution in all solution space.)

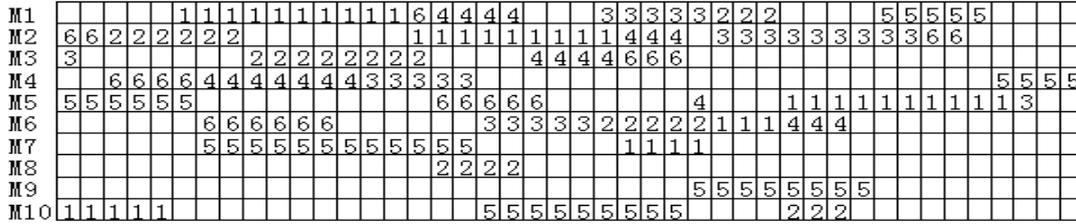


Figure 1 Gantt chart of the smaller total running time of the machine, $f_1 = 44 (f_2 = 136)$

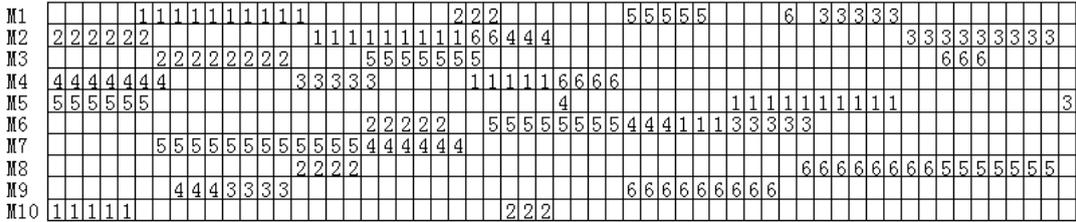


Figure 2 Gantt chart of the smaller machine transfer costs, $f_2 = 90 (f_1 = 59)$

Experiment 1 focuses on the minimum solution of f_1 . Experiment 2 focuses on the minimum solution of f_2 . From experiment 1 and experiment 2, we can see that f_1 is smaller, f_2 is larger, f_2 is smaller, f_1 is larger. The purpose of experiment 3 is to find a more balanced solution, so that f_1 and f_2 have reached better solution, that is f_1 close to 44, f_2 close to 90, which was called a good solution.

4. Design of Solving Multi - Objective Problem based on Bucket Principle and Genetic Algorithm

4.1 Algorithm flow chart

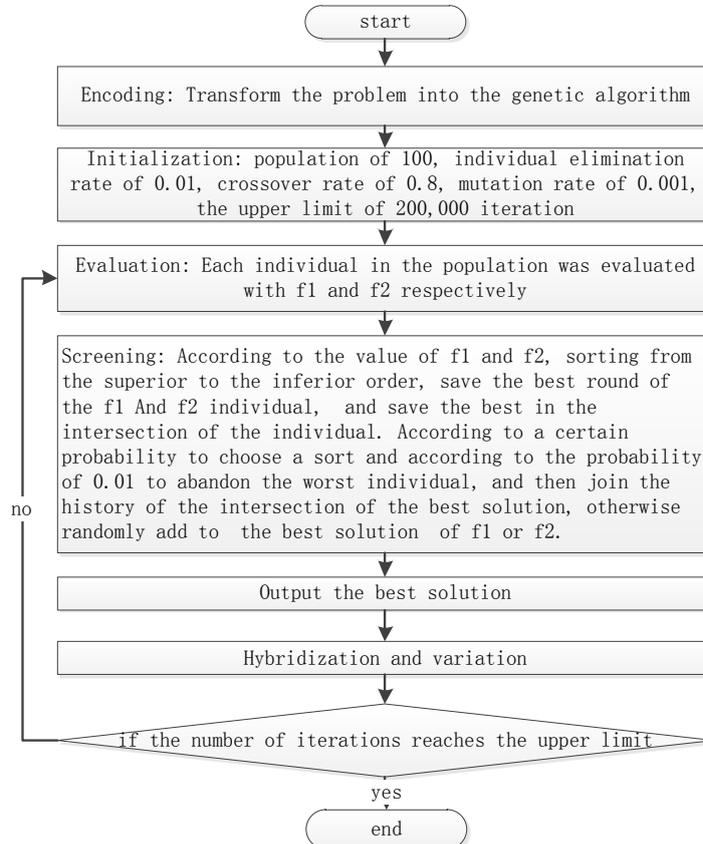


Figure 3 Algorithm flow chart based on bucket principle and genetic algorithm

4.2 Detailed steps

1) Encoding

Encoding method use two-dimensional array, the processing sequence coding method take 6 parts of the six processes as six teams, putting them into a team. The merger method is randomly selected a column from the rest of the team, so the row of the column can entry into the new team, until you get a length of 36 for the process sequence as coding. When the processing sequence is determined, the machine part of the process is randomly selected from the optional machine. Getting code of the individual as shown in Table 4. The first column is explained: giving priority to the first step of the first part, the first step processing in the third machine, processing time is 3, and so on.

Table 4 encoding method

Sequence of parts	1	4	6	3	2	6	5	5	1	3	2	6	4	1	...
Machine selection	3	4	2	9	2	7	5	2	1	7	3	9	9	2	...
Processing time	3	7	2	4	6	7	6	10	10	7	8	9	3	9	...

2) Initialize

Population size, individual elimination rate, crossover rate, mutation rate should be set to be reasonable, otherwise, then it may lead to excessive population convergence and premature loss of searching ability, or lead to too slow or not convergence, which get into a completely random search situation. Here the search limit is set to 200000 is relatively large, because the solution space is too large reasons, other values set in Figure 3.

3) Evaluation

The evaluation mechanism of f_1 take the the way of simulation. According to the processing sequence in the code to arrange the parts on the machine for processing, statistical processing time, which is presented by Gantt chart. Gantt chart horizontal lattice number is the total length of processing time; the evaluation mechanism of f_2 only needs encode the individual machine, and each part needs 6 steps. Therefore, each part is transferred 5 times, the number of all parts transferred is 30 times, and the cost of each transfer is accumulated, which is the value of f_2 . The results listed in Figure 1 are shown in Table 3:

Transfer process of Part 1: M10->M1->M2->M7->M6->M5, cost: 1-2-6-8-6

Transfer process of Part2: M2->M3->M8->M6->M1->M10, cost: 8-1-5-1-1

Transfer process of Part3: M3->M4->M6->M1->M2->M5, cost: 2-4-1-2-6

Transfer process of Part4: M4->M1->M3->M2->M5->M6, cost: 7-9-8-6-6

Transfer process of Part5: M5->M7->M10->M9->M1->M4, cost: 4-4-9-2-7

Transfer process of Part6: M2->M4->M6->M1->M5->M3, cost: 2-4-1-5-8

$$f_2 = 1 + 2 + 6 + \dots + 1 + 5 + 8 = 136$$

4) Screening

This step fully embodies the role of the principle of the barrel. The screening process is to sort the population according to the value of the two objective functions respectively, sort from best to worst. For both sorting, randomly select a sort, and according to the probability of 0.01, the sort of the worst one that is eliminated, which is also known as the removal of short board. In this process, in order to study the problem, trying out a different knockout rate for both sorties respectively, observe Inge effect of the good solution. When we phase out, the number of populations is bound to decrease, in order to maintain the number of groups unchanged, it is necessary to restore the number of populations, you need to add to the population, this supplement has three sources: the first is the search history of the best f_1 individuals, the second is to search the history of the best f_2 individuals, the third is to search the history of the two sort of the best individual in the intersection. The elements of this intersection are the main search objects of this problem. Theoretically, this intersection may not be empty in each iteration, or it may be absent only in some round iterations, or in all rounds iterations. When this intersection is present, we take it as the main source of supplement, and if it does not exist, the random selection of f_1 or f_2 in the search history is supplemented. Thus, it is necessary to determine whether or not these three

solutions are better in each iteration. When there is a better solution, update the search history of these three kinds of solutions to prepare for the next round of iteration.

5) Hybridization and mutation methods

For two individuals, a hybrid site was randomly selected, the gene of one of the individuals was retained, and the same gene of the second individual was removed, and the two genes were recombined into one gene. The gene was used to describe the hybridization process as follows.

Assuming the genetic fragments of the two parents are:

individual 1	1	2	4	2	3	4	5	6
individual 2	3	4	6	5	2	2	1	4

Randomly selecting a hybrid position, such as position = 3, according to this position for gene retention and gene knockout

individual 1	retaine4			4			4	
individual 2	remove4	3		6	5	2	2	1

Synthesizing a new individual

New individual 1		3	6	4	5	2	4	2	1
------------------	--	---	---	---	---	---	---	---	---

And then removed and retained in turn

individual 1	remove6		1	2	4	2	3	4	5
individual 2	retaine6				6				

Synthesizing another new individual

New individual 2		1	2	6	4	2	3	4	5
------------------	--	---	---	---	---	---	---	---	---

The method of mutation: For each individual, randomly selected a location in the parts processing sequence table, then a part must be selected. To extract all the genes of this part, including parts processing order, machine selection and processing time. And then randomly inserting into the team inside, keeping the order of 6 procedures do not change when inserting. The method of inserting as follows: randomly selecting the location of the team after the extraction of the gene, inserting the first process, and then randomly inserting the second process in the first process, and inserting the rest of the process, until the six processes completely back into the team, and get the length of 36 teams. This new gene is not the same as the original with a large probability, which can achieve the purpose of mutation.

4.3 Comparing and analyzing simulation experiment data

The following table listed the results based on the principle of barrel and genetic algorithm.

group	The first group				The second group				The third group			
Out rate	f1:0.9		f2:0.1		f1:0.5		f2:0.5		f1:0.1		f2:0.9	
value	f1	f2	p	g	f1	f2	p	g	f1	f2	p	g
1	78	116	3	1	78	116	3	1	78	116	3	1
2	74	110	1	199	73	116	1	294	76	116	1	113
3	69	110	2	914	72	116	3	297	70	113	3	460
4	67	110	1	916	66	111	1	714	68	113	4	461
5	66	110	3	1375	66	109	2	2025	68	108	1	487
6
7	51	91	1	56960	50	95	6	61590	53	90	2	112316
8	49	91	1	77440	50	94	3	88757	52	90	1	116720
9	48	91	1	81764	48	94	2	103112	51	90	3	117463
10	47	91	4	93885	47	94	1	103115	50	90	3	120170
11	46	91	5	121558	47	92	2	118193	47	90	6	122566

The experiment is divided into three groups, f1 is the value of the objective function f1, f2 is the value of the objective function f2, p represents the number of intersection elements in the order of f1 and f2, g represents the solution is the iteration in g. Due to the limitation of space, the first five solutions and the last five solutions in the iteration process are listed for each set of experiments. From this table there are three discoveries: one, from the vertical direction of the table, with the increase in the number of iterations, the solution is getting better and better, and the emergence of the solution was spiral progressive change, that is, the solution of the two components roughly close to the optimal value of the single objective function. The direction of the change is in accordance with the solution of

the Pareto solution, that is, the scheme tries to be better and if there is no better solution, the solution is accepted. Two, the two sorting elements of the intersection do not have so much in each iteration process, the two objective functions has inherent conflict. Three, from the table horizontal view, for the function out of the set rate, which function is higher, the solution is more conducive to this function, but no more data to support this conclusion. In short, no matter how much set out the elimination rate, it can find the good solution, from the three groups of experiments, three ideal solutions were found :(46, 91), (47, 92), (47, 90), their first component is very close to the optimal solution of the single objective function f_1 , $f_1 = 44$, and the second component is close to the optimal solution of the single objective function f_2 , $f_2 = 90$.

5. Conclusion

As described above, under the background of 2 targets, this paper attempts to solve the multi-objective problem by the genetic algorithm based on the bucket principle, which showing that it is a very effective algorithm, its programming idea is clear and the process is relatively simple. In the presence of the optimal solution, it is possible to ensure that the optimal solution is found and that at least one Pareto optimal solution can be found. In the case of solving the practical problem, it is completely acceptable when the solution is not demanding, because the solution of each objective function values are close to the minimum. Based on the bucket principle and the genetic algorithm, the process of solving the optimal solution is to simplified. There is no need to use too complicated techniques, only need to simply remove the worst. This algorithm not only can solve the practical problems in production, but also maintain the original multi-objective situation, which the multi-objective problem is not converted into a single target problem, and the solution is easy to understand and accept. In addition, this algorithm is inventive, combining the simple principle - the bucket principle and modern optimization algorithm, which are worthy for people studying in-depth from the theory and practice.

6. Acknowledgements

This work was financially supported by National Natural Science Foundation of China (NSFC) (61572498).

7. References

- [1]. Xu Yuanbin. Optimization Model of Large Data Energy Saving Based on Genetic Algorithm [J]. Computer and Modernization, 2016, (10):25.
- [2]. He Yuan, Zhang Yuqing, Zhang Guanghua. Excavation of Android Driver Vulnerability Based on Black Box Genetic Algorithm [J]. Journal of Computer Science, 2016,:1-14.
- [3]. He Yi-chao, Wang Xizhao, Li Wenbin, Zhang Xinlu, Chen Yiying. Research on Solving Discount {0-1} Knapsack Problem Based on Genetic Algorithm [J]. Journal of Computer Science, 2016, (12):2614-2630.
- [4]. Qi Yutao, Liu Fang, Ren Yuan, Liu Jingle, Jiao Li Cheng. Cooperative Immune Multi - objective Optimization Algorithm Based on Cooperative Model [J]. Journal of Electronics, 2014, (05):858-867.
- [5]. Lin Chuang, Chen Ying, Huang Jiwei, Xiang Xudong. Multi - objective Optimization Model and Solving of Service Quality in Service Calculation [J]. Journal of Computer Science, 2015, (10):1907-1923.
- [6]. Chen Baichao, Wei Liangliang, Lei Yang, Tu Zhikang, Yuan Jiabin, Chen Feng, Nie Dexin, Pan Jing. Pareto multi - objective optimal configuration of high temperature superconducting fault current limiter based on immune algorithm [J]. Power grid technology, 2015, (05):1343-1350.
- [7]. Xiao Jing, Bi Xiaojun, Wang Kejun. Research on High - dimensional Multi - objective Optimization Based on Global Ranking [J]. Journal of Software, 2015, (07):1574-1583.

- [8]. Wang Juan, Xu Fu Xia. Interval Estimation of Mesoscale Parameters in Pareto Distribution [J]. *Journal of Harbin University of Commerce (Natural Science Edition)*, 2015, (05):629-633.
- [9]. Liu Rongxuan, Wu Siying, Zhang Ping. Parameter Estimation of Three - parameter Pareto Distribution [J]. *Journal of Jinggangshan University (Natural Science Edition)*, 2015, (04):7-12.
- [10]. Zhang Min. The use of tradable electronic way ticket to achieve the traffic network Pareto improvement [D]. Southwest Jiaotong University, 2015.
- [11]. Peng Zongping. "Short-board Principle" and Overseas Chinese Learning - A Case Study of Chinese-American Mixed Chinese Adult Chinese Learners [J]. *Education and Teaching Forum*, 2014, (53):85-86.
- [12]. Xie Xuan Yang. On the Management Research Method from the Development History of the Principle of Barral [J]. *Research on Technology Economy and Management*, 2013, (04):50-54.
- [13]. Li Qing, Zhang Yinxia, Zhang Yanmiao, Liu Xiaoyin, Mao Li Mei. Application of "cask principle" to improve the quality management of automatic tablet dispenser [J]. *Chinese Journal of Hospital Pharmacy*, 2015, (07):648-650.